HOW SENSITIVE ARE META-LEARNERS TO DATASET IMBALANCE?

Anonymous authors
Paper under double-blind review

ABSTRACT

Meta-Learning (ML) has proven to be a useful tool for training Few-Shot Learning (FSL) algorithms by exposure to batches of tasks sampled from a meta-dataset. However, the standard training procedure overlooks the dynamic nature of the real-world where object classes are likely to occur at different frequencies. While it is generally understood that imbalanced tasks harm the performance of supervised methods, there is no significant research examining the impact of imbalanced meta-datasets on the FSL evaluation task. This study exposes the magnitude and extent of this problem. Our results show that ML methods are more robust against meta-dataset imbalance than imbalance at the task-level with a similar imbalance ratio ($\rho < 20$), with the effect holding even in long-tail datasets under a larger imbalance ($\rho = 65$). Overall, these results highlight an implicit strength of ML algorithms, capable of learning generalizable features under dataset imbalance and domain shift.

1 INTRODUCTION

Few-Shot Learning (FSL) aims at reducing the burden of training machine-learning models on a large number of labeled data points. A common way of training FSL models is through Meta-Learning (ML) (Hospedales et al., 2020; Vinyals et al., 2017) with the model repeatedly exposed to batches of FSL tasks/episodes sampled from a (meta-)training dataset that is different but similar to the one seen during the (meta-)testing phase. We will use the prefix *"meta"* to distinguish the high-level training and evaluation routines of ML (outer loop) from the training and evaluation routines at the single-task level (inner loop).

Motivation. Standard FSL benchmarks (eg. Omniglot, Mini-ImageNet) assume an equal number of data-points for each class in the meta-dataset. However, in real-world applications it is common to encounter imbalanced datasets with different classes having a different number of data-points (Guan et al., 2020; Ochal et al., 2020). In some cases, the dataset distribution could be immeasurably large or even unknown, like in online class-imbalance (Wang et al., 2014). Although meta-dataset imbalance is present in recent FSL benchmarks, such as Meta-Dataset (Triantafillou et al., 2020) or meta-iNat (Wertheimer & Hariharan, 2019), its impact on ML has remained unexplored.

Contributions. We provide quantitative insights into the dataset imbalance problem and show that meta-learners are robust to meta-dataset imbalance under 1) various imbalance distributions, 2) dataset sizes, and 3) moderate cross-domain shift. The impact of imbalance at task level is much more significant in comparison.

2 RELATED WORK

Class Imbalance. In classification, imbalance occurs when at least one class (the majority class) contains a higher number of samples than the others. The classes with the lowest number of samples are called minority classes. If uncorrected, conventional supervised loss functions, such as (multi-class) cross-entropy, skew the learning process in favor of the majority class, introducing bias and poor generalization toward the minority class samples (Buda et al., 2018; Leevy et al., 2018). The object recognition community studies class imbalance using real-world datasets or distributions that approximate real-world imbalance (Buda et al., 2018; Johnson & Khoshgoftaar, 2019; Liu et al., 2019). Buda et al. (2018) state that two distributions can be used: *linear* and *step* imbalance. At large-scale, datasets with many samples and classes tend to follow a *long-tail* distribution (Liu et al., 2019; Salakhutdinov et al., 2011; Reed, 2001), with most of the classes occurring with small frequency and a few classes occurring with high frequency.

Meta-Dataset FSL Benchmarks. FSL methods (Snell et al., 2017; Sung et al., 2017; Edwards & Storkey, 2017; Vinyals et al., 2017; Finn et al., 2017; Ravi & Larochelle, 2016; Chen et al., 2019; Dhillon et al., 2020; Patacchiola et al., 2020; Zhang et al., 2021) are typically compared against balanced benchmark (eg. Omniglot, MiniImageNet). More recent benchmarks (Triantafillou et al., 2020; Wertheimer & Hariharan, 2019) contain some levels of imbalance in the meta-dataset. Specifically, Meta-Dataset (Triantafillou et al., 2020) combines datasets of different size (e.g. Omniglot, CUB, Fungi, Aircraft, ImageNet etc.) into a single corpus. A different dataset, meta-iNat (Wertheimer & Hariharan, 2019), models imbalance according to a long-tail distribution. While imbalance at the *task-level* has received some attention (Triantafillou et al., 2020; Chen et al., 2020; Lee et al., 2019; Guan et al., 2020), limited research quantifies the impact of imbalance at the *dataset-level*. It can be cumbersome to evaluate the imbalance of Meta-Dataset and meta-iNat specifically, as it requires access to a balanced version of the datasets with an equal total number of samples. Therefore, in our analysis we artificially induce imbalance into datasets, and model imbalance according to various distributions approximating many real-world scenarios.

3 Problem Definition

Standard Task/Meta-Training. Benchmarking FSL methods typically involves three phases: meta-training/pre-training, meta-validation, and meta-testing. Each phase samples batches of data points or tasks from a separate dataset: $\mathcal{D}_{train}$, $\mathcal{D}_{val}$, and $\mathcal{D}_{test}$, respectively, such that the classes and samples between each of the datasets are non-overlapping. We assume that a dataset $\mathcal{D}$ is $balanced$ when it contains $N^{\mathcal{D}}$ classes and $K^{\mathcal{D}}$ samples for each class. Similarly, a standard $K^{\mathcal{S}}$-shot $N^{\mathcal{S}}$-way FSL classification task is defined by a small *support set*, $\mathcal{S} = \{(x_1, y_1), \ldots, (x_s, y_s)\} \sim \mathcal{D}$, containing $N^{\mathcal{S}} \times K^{\mathcal{S}}$ image-label pairs drawn from $N^{\mathcal{S}}$ unique classes with $K^{\mathcal{S}}$ samples per class ($|\mathcal{S}| = K^{\mathcal{S}} \times N^{\mathcal{S}}$ and $K^{\mathcal{S}} \ll K^{\mathcal{D}}$ and $N^{\mathcal{S}} \ll N^{\mathcal{D}}$). The goal is to minimize some loss over a *query set*, $\mathcal{Q} = \{(x_1, y_1), \ldots, (x_q, y_q)\} \sim \mathcal{D}$, containing a different set of $K^{\mathcal{Q}}$ samples drawn from the same $N^{\mathcal{S}}$ classes (i.e. $N^{\mathcal{Q}} = N^{\mathcal{S}}$, $\mathcal{Q}^{(x)} \cap \mathcal{S}^{(x)} = \emptyset$ and $\mathcal{Q}^{(y)} \equiv \mathcal{S}^{(y)}$).

Imbalanced Tasks/Meta-Dataset. For brevity, but without loss of generality, we define a distribution for a set of data points $* \in \{\mathcal{D}, \mathcal{S}, \mathcal{Q}\}$ as a tuple (K^*_{min}, K^*_{max}, N^*, M^*) for a distribution $\mathcal{I} \in \{linear, step, long\text{-}tail\}$ (Buda et al., 2018; Wertheimer & Hariharan, 2019), where K^*_{min} is the minimum number of samples per class, K^*_{max} is the maximum number of samples per class, N^* is the number of classes, and M^* is an additional parameter used for $step$ and $long\text{-}tail$ imbalance. Specifically, we define $\mathcal{I}$ distributions as:

- *Linear imbalance*. The K^*_i number of samples for each class $i \in \{1..N^*\}$ is defined by:

$$K^*_i = \texttt{round}(K^*_{min} - c + (i-1) \times (K^*_{max} + 2 \times c - K^*_{min})/(N^* - 1)), \quad (1)$$

  where $c = 0.499$ for rounding purposes. For example, this means that for $linear$ (1,9,5,-) set, $K^*_i \in \{1, 3, 5, 7, 9\}$, and for $linear$ (4,6,5,-) set, $K^*_i \in \{4, 4, 5, 6, 6\}$.

- *Step imbalance*. The number of class samples, K^*_i, is determined by an additional variable M^* specifying the number of minority classes. Specifically, for classes $i \in \{1..N^*\}$:

$$K^*_i = \begin{cases} K^*_{min}, & \text{if } i \leq M^*, \\ K^*_{max}, & \text{otherwise.} \end{cases} \quad (2)$$

  For example, in a $step$ (1,9,5,1) set, there is 1 minority class, and $K^*_i \in \{1, 9, 9, 9, 9\}$.

- *Long-Tail imbalance*. Imbalance could be modeled by a Zipf's/Power Law (Reed, 2001) for a more realistic imbalance distribution.

We also report the imbalance ratio ρ, which is a scalar identifying the level of class-imbalance; this is often reported in the CI literature for the supervised case (Buda et al., 2018). We define ρ to be the ratio between the number of samples in the majority and minority classes in a set of data points:

$$\rho = \frac{K^*_{max}}{K^*_{min}}. \quad (3)$$

4 Experiments

Setup We meta-trained a range of FSL and ML methods on multiple datasets and imbalance distributions. Specifically, we investigate Prototypical Networks (Snell et al., 2017), Matching Networks (Vinyals et al., 2017), Relation Networks (Sung et al., 2017), MAML (Finn et al., 2017),

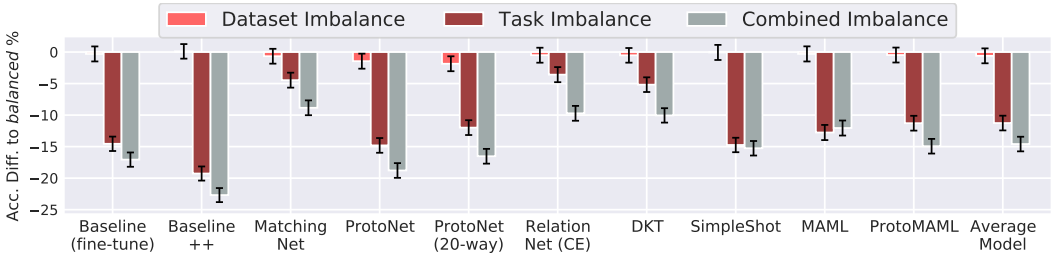

Figure 1: Difference in accuracy between dataset imbalance (light red), task imbalance (dark red), and combined imbalance (grey) against the $balanced$ meta-training baseline. Negative performance indicates a lower accuracy compared to $balanced$. The results suggest that models are quite robust to dataset imbalance compared to imbalance at the task-level.

ProtoMAML (Triantafillou et al., 2020), DKT (Patacchiola et al., 2020), SimpleShot (Wang et al., 2019), and supervised pre-training methods – Baseline and Baseline++ from Chen et al. (2019). Each model was meta-trained/pre-trained on 100k tasks/mini-batches sampled from variations of the meta-training dataset of Mini-ImageNet (Ravi & Larochelle, 2016) ($\mathcal{D}_{train}$), originally containing 64 classes and 600 samples per class. To get enough samples for the majority classes, we halved the total number of samples from the original $\mathcal{D}_{train}$ of Mini-ImageNet, and denote this dataset as $\mathcal{D}'_{train}$, where $|\mathcal{D}'_{train}| \approx 64 \times 300 = 19200$. We induced imbalance into the dataset according to one of the $\mathcal{I}$-distributions outlined in Section 3. For simplicity, we did not modify the meta-validation and meta-testing datasets ($\mathcal{D}_{val}$ and $\mathcal{D}_{test}$), keeping them like in the original Mini-ImageNet. To emulate stronger domain-shift, we also evaluated models on 50 randomly selected classes from CUB-2011 (Wah et al., 2011). More details and experiments are in Appendix A.

Meta-learners are robust to dataset imbalance. We compare dataset imbalance to task imbalance using a similar distribution. In the *dataset imbalance* condition, tasks (5-shot, 5-way) are sampled from a a linearly-imbalanced dataset $\mathcal{D}'_{train}$ (30,570,64,-). In the *task imbalance* condition, tasks are sampled from a linearly-imbalanced distribution (1,9,5,-) – also called 1-9shot 5-way – and from a balanced dataset $\mathcal{D}'_{train}$ (300,300,64,-). Finally, in the *combined imbalance* condition imbalanced tasks (1,9,5,-) are sampled from the imbalanced dataset $\mathcal{D}'_{train}$ (30,570,64,-). Figure 1 shows the difference in $\mathcal{D}_{test}$ accuracy between dataset imbalance (light red) and task imbalance (dark red) compared against the score of standard training ($balanced$ at both dataset and task levels). Dataset imbalance causes an insignificant drop in performance, being mostly within error-bars with respect to $balanced$. On the other hand, the imbalanced task condition causes a significant drop in performance, up to -20% for some methods despite the slightly smaller imbalance magnitude ($\rho = 9$ c.f. $\rho = 19$). This result suggests that meta-learners are quite robust to dataset imbalance. Interestingly, the combined task and dataset imbalance seems to have a slight compounding effect on performance, yielding -15% on average, and performing worse than the combined sum of the two imbalance conditions taken individually (-12%).

Table 1: Evaluation accuracy after meta-training on $\mathcal{D}'_{train}$ derived from Mini-ImageNet with various imbalance distributions. Small differences in accuracy between $balanced$ and other distributions, suggest a small effect of imbalance at dataset level. *Left:* Evaluation on the meta-testing dataset of Mini-ImageNet. *Right:* Evaluation on the meta-testing dataset of CUB.

$\mathcal{D}'_{train} \rightarrow \mathcal{D}_{test}$	Mini-ImageNet → Mini-ImageNet				Mini-ImageNet → CUB			
Imbalance in $\mathcal{D}'_{train}$	*balanced* (300, 300, 64, -)	*step*-32 (30, 570, 64, 32)	*step*-22 (25, 444, 64, 22)	*step*-animal (25, 444, 64, 22)	*balanced* (300, 300, 64, -)	*step*-32 (30, 570, 64, 32)	*step*-22 (25, 444, 64, 22)	*step*-animal (25, 444, 64, 22)
Baseline (fine-tune)	61.23$\pm$0.72	60.44$\pm$0.71	60.36$\pm$0.71	59.32$\pm$0.73	56.61$\pm$0.72	55.83$\pm$0.71	56.35$\pm$0.72	52.83$\pm$0.73
Baseline++	63.86$\pm$0.67	**63.23$\pm$0.66**	**62.91$\pm$0.67**	62.03$\pm$0.70	57.71$\pm$0.72	56.62$\pm$0.75	**57.07$\pm$0.75**	52.07$\pm$0.71
Matching Net	62.05$\pm$0.69	58.23$\pm$0.69	58.82$\pm$0.70	59.17$\pm$0.71	52.64$\pm$0.74	51.10$\pm$0.74	50.20$\pm$0.75	50.78$\pm$0.75
ProtoNet	64.00$\pm$0.71	60.40$\pm$0.70	60.52$\pm$0.70	61.22$\pm$0.70	54.33$\pm$0.73	52.19$\pm$0.73	52.78$\pm$0.73	52.34$\pm$0.73
ProtoNet (20-way)	**65.41$\pm$0.70**	59.75$\pm$0.71	60.47$\pm$0.70	61.17$\pm$0.72	55.46$\pm$0.74	52.73$\pm$0.73	52.97$\pm$0.71	50.56$\pm$0.72
Relation Net (CE)	63.78$\pm$0.70	58.15$\pm$0.70	60.50$\pm$0.69	59.66$\pm$0.70	55.83$\pm$0.73	52.90$\pm$0.69	52.87$\pm$0.72	52.35$\pm$0.73
DKT	62.31$\pm$0.67	59.01$\pm$0.68	59.27$\pm$0.69	59.82$\pm$0.67	**58.01$\pm$0.74**	**56.97$\pm$0.73**	56.72$\pm$0.73	**56.26$\pm$0.72**
SimpleShot	62.53$\pm$0.71	62.34$\pm$0.72	62.03$\pm$0.71	61.26$\pm$0.75	56.64$\pm$0.74	55.54$\pm$0.73	56.01$\pm$0.73	52.43$\pm$0.72
MAML	61.20$\pm$0.72	58.29$\pm$0.73	58.97$\pm$0.71	59.36$\pm$0.71	55.62$\pm$0.74	55.13$\pm$0.74	54.50$\pm$0.73	54.57$\pm$0.74
ProtoMAML	64.78$\pm$0.70	60.67$\pm$0.71	61.77$\pm$0.71	**62.41$\pm$0.71**	57.02$\pm$0.74	55.32$\pm$0.73	55.82$\pm$0.71	54.88$\pm$0.73
Avr. Diff. to *balanced*	-	-3.1	-2.6	-2.6	-	-1.6	-1.5	-3.1

Robustness against domain-shift. For the remaining sections, we kept the tasks balanced (5-shot 5-way), digging deeper into dataset imbalance. In this paragraph, we examined various $step$ imbalance distributions and evaluated performance with domain-shift. Table 1 shows the meta-testing performance on $\mathcal{D}_{test}$ of Mini-ImageNet (left column set) or CUB (right column set) after training on $step$ imbalanced $\mathcal{D}'_{train}$. The bottom row shows the average model difference in accuracy between $balanced$ and the imbalanced datasets. The results show a small negative difference (-3.1%) between the $balanced$ dataset and the step scenario with 32 minority classes ($step$-32). Evaluation of models trained under the $step$-32 distribution on CUB, suggests only a small difference (-1.6) under the domain-shift condition. Note that, D'_{train} contains 22 animal classes including 3 classes of birds and this might not represent a very large domain-shift. In Figure 2 and Appendix C.1, we examine more $step$ imbalance distributions on a smaller dataset containing only 1/8th of total samples of $\mathcal{D}_{train}$ from Mini-ImageNet. Overall, the performance drops as the number of minority classes increases but the effect still remains quite low ($< 3\%$ absolute difference).

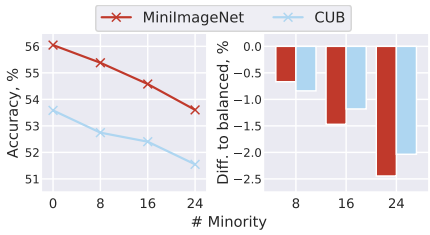

Figure 2: Combined model average performance with increasing minority classes. *Left:* Combined accuracy of all models. *Right:* Performance difference to the balanced dataset.

Robustness against larger domain-shift. Next, we examined a stronger domain-shift scenario. We set all 22 animal classes in $\mathcal{D}'_{train}$ to contain $K^{\mathcal{D}}_{min} = 25$ samples each, and we set the other 42 classes to have $K^{\mathcal{D}}_{max} = 444$ samples each. We call this setting "$step$-animal" as it reduces the diversity of animal samples seen during meta-training. As a control variable, we examine a similar $step$ imbalance with 22 minority classes picked uniformly at random ($step$-22). Results in Table 1 suggest that $step$-animal performs slightly worse (1.6%) compared to $step$-22 on average. SimpleShot and the two Baselines are particularly affected by the larger domain-shift on CUB, getting -4.0% performance drop on $step$-animal compared to $step$-22. Perhaps, this drop is due to the particular training procedure used for these methods which are pre-trained on mini-batches instead of tasks. This suggests an implicit strength of ML algorithms against larger domain-shift.

Robustness against larger imbalance. In Appendix C.2, we examined $long\text{-}tail$ distributions with an imbalance ratio $\rho = 65$ on a larger dataset ($6.5\times$ larger than $\mathcal{D}'_{train}$, and derived directly from ImageNet). We observed a more significant drop in accuracy w.r.t. the balanced dataset for ProtoNet and MAML (-6.3% and -4.1%, respectively). This suggests that a more natural imbalance distribution and a higher imbalance ratio could cause a larger performance drop. However, the performance drop still remained smaller compared to task-level imbalance in Figure 1.

5 Discussion and Conclusion

In this work, we have provided insights into the meta-dataset imbalance problem in meta-learning, showing that models are quite robust to meta-dataset imbalance – meaning that they would likely experience only small drops in accuracy when exposed to dataset imbalance in the real-world. In contrast, the support-set imbalance yields significantly larger (an order of magnitude) performance drop. Overall, our results suggest that dataset imbalance has a small negative effect on the ML procedure when tasks are balanced. This seems to point to an implicit strength of ML algorithms, which are able to learn generalizable features if exposed to balanced tasks during the inner-loop optimization. Higher levels of step and long-tail imbalance will likely produce more dramatic performance differences. In this study, we evaluated generalization to novel tasks and classes only - however, it is likely that imbalance in $\mathcal{D}_{train}$ would impact performance on base tasks and classes sampled from D_{train}. Moreover, in this study, we were constrained by the size of meta-training dataset of Mini-ImageNet and ImageNet as we had to control the size of the dataset. In the real-world, meta-datasets can be very large with very high imbalance ratios ($\rho \gg 100$) and following long-tail distributions (Liu et al., 2019; Salakhutdinov et al., 2011). Future work should investigate how these larger imbalance level could affect the meta-learning ability and cross-domain generalization.

REFERENCES

Antreas Antoniou, Massimiliano Patacchiola, Mateusz Ochal, and Amos Storkey. Defining Benchmarks for Continual Few-Shot Learning. *arXiv preprint arXiv:2004.11967*, 2020.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Wei Yu Chen, Yu Chiang Frank Wang, Yen Cheng Liu, Zsolt Kira, and Jia Bin Huang. A closer look at few-shot classification. *International Conference on Learning Representations (ICLR)*, 2019.

Xinshi Chen, Hanjun Dai, Yu Li, Xin Gao, and Le Song. Learning to Stop While Learning to Predict. *International Conference on Machine Learning (ICML)*, 2020.

Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A Baseline for Few-Shot Image Classification. In *International Conference on Learning Representations*, 2020.

Harrison Edwards and Amos Storkey. Towards a Neural Statistician. *International Conference on Learning Representations (ICLR)*, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *International Conference on Machine Learning (ICML)*, 2017.

Jian Guan, Jiabei Liu, Jianguo Sun, Pengming Feng, Tong Shuai, and Wenwu Wang. Meta Metric Learning for Highly Imbalanced Aerial Scene Classification. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4047–4051. IEEE, 2020.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-Learning in Neural Networks: A Survey. *arXiv preprint arXiv:2004.05439*, 2020.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning (ICML)*, 2015.

Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6, 2019.

Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to Balance: Bayesian Meta-Learning for Imbalanced and Out-of-distribution Tasks. *International Conference on Machine Learning (ICML)*, 2019.

Joffrey L. Leevy, Taghi M. Khoshgoftaar, Richard A. Bauder, and Naeem Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5, 2018.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-Scale Long-Tailed Recognition in an Open World. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2537–2546, 2019.

Mateusz Ochal, Jose Vazquez, Yvan Petillot, and Sen Wang. A Comparison of Few-Shot Learning Methods for Underwater Optical and Sonar Image Classification. *OCEANS 2020 preprint*, 2020.

Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

Massimiliano Patacchiola, Jack Turner, Elliot J. Crowley, Michael O'Boyle, and Amos Storkey. Bayesian Meta-Learning in the Few-Shot Setting via Deep Kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations (ICLR)*, 2016.

William J Reed. The pareto, zipf and other power laws. *Economics Letters*, 2001.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Jake Snell and Richard Zemel. Bayesian Few-Shot Classification with One-vs-Each Polya-Gamma Augmented Gaussian Processes. *arXiv preprint arXiv:2007.10417*, 2020.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical Networks for Few-shot Learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to Compare: Relation Network for Few-Shot Learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples. *International Conference on Learning Representations (ICLR)*, 2020.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Evan Vogelbaum, Rumen Dangovski, Li Jing, and Marin Soljačić. Contextualizing Enhances Gradient Based Meta Learning. *arXiv preprint arXiv:2007.10143*, 2020.

C. Wah, S Branson, P Welinder, P Perona, and S Belongie. The Caltech-UCSD Birds-200-2011 Dataset. In *California Institute of Technology*, 2011.

Shuo Wang, Leandro L Minku, and Xin Yao. Resampling-based ensemble methods for online class imbalance learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1356–1368, 2014.

Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning. *arXiv preprint arXiv:1911.04623*, 2019.

Davis Wertheimer and Bharath Hariharan. Few-Shot Learning with Localization in Realistic Settings. *arXiv preprint arXiv:1904.08502*, pp. 6558–6567, 2019.

Xueting Zhang, Debin Meng, Henry Gouk, and Timothy Hospedales. Shallow bayesian meta learning for real-world few-shot recognition. *arXiv preprint arXiv:2101.02833*, 2021.

A IMPLEMENTATION DETAILS

A.1 FSL METHODS AND BASELINES

In our experiments, we examined a wide range of meta-learning and few-shot learning models and baselines. Hyper-parameters are located in our source code.

1. **Baseline (fine-tune)** (Pan & Yang, 2010; Chen et al., 2019) represents a classical way of applying transfer learning, where a neural network is pre-trained on a large dataset, then fine-tuned on a smaller domain-specific dataset. The backbone of baseline has a single linear classification layer with an output for each meta-training class. The network was trained during pre-training. During meta-testing, the pre-trained linear layer was exchanged for another randomly initialized layer with outputs matching the number of classes in the tasks (N-way). Fine-tuning was performed on the new randomly initialized classification layer using the support set $\mathcal{S}$.
2. **Baseline++** (Chen et al., 2019) augments the fine-tune baseline by using Cosine Similarity on the last layer.
3. **Matching Network (Matching Net)** (Vinyals et al., 2017) uses context embeddings with an LSTM to effectively perform k-nearest neighbor in embedding space using cosine similarity to classify the query set.
4. **Prototypical Networks (ProtoNet)** (Snell et al., 2017) maps images into a feature space and calculates class means (called prototypes). The query samples are then classified based on the closest Euclidian distance to class prototypes. We evaluate two models, the first meta-trained like the others on 5-way episodes, and the second trained on 20-way episodes. During 20-way meta-training, we set the the query size to 5.
5. **Relation Networks (Relation Net)** (Sung et al., 2017) augment the classical Prototypical Networks by introducing a relation module (another neural network) that compares the distance using a learnable score. The original method uses Mean Squared Error to minimize the relation score between samples of the same type. However, we follow Chen et al. (2019) and use cross-entropy loss to improve performances. The structure of the relation module is described in section A.2.
6. **DKT** (formally called GPShot) proposed by Patacchiola et al. (2020) is a probabilistic approach that utilizes the Gaussian Processes with a deep neural network as a kernel function. We used Batch Norm Cosine distance for the kernel type.
7. **SimpleShot** (Wang et al., 2019) augments the 1-NN baseline model by normalizing and centering the feature vector using the mean feature vector of the dataset. The query samples are assigned to the nearest prototype according to the Euclidian distance. In contrast to the baseline models, pre-training is performed on the meta-training dataset like other meta-learning algorithms, and meta-validation is used to select the best model based on on tasks sampled from $\mathcal{D}_{val}$.
8. **MAML** (Finn et al., 2017) is a meta-learning technique that learns a common initialization of weights that can be quickly adapted for various tasks using fine-tuning on the support set. The task adaptation uses a standard gradient descent algorithm minimizing Cross-Entropy loss on the support set. The original method uses second-order derivates; however, due to more efficient calculation, we use the first-order MAML, which has been shown to work just as well. We set the inner-learning rate to 0.1 with 10 iteration steps. We optimize the meta-learner model on batches of 4 meta-training tasks. These hyperparameters were selected based on our hyperparameter fine-tuning.
9. **ProtoMAML** (Triantafillou et al., 2020) augments traditional first-order MAML by reinitializing the last classification layer between tasks. Specifically, the weights of the layer are assigned to the prototype for each corresponding output. This extra step combines the fine-tuning ability of MAML and the class regularization ability of Prototypical Networks. We set the inner-loop learning rate to 0.01 with 5 iterations. Unlike MAML, we found that updating the meta-learner after a single meta-training task gave the best performance.

A.2 BACKBONE ARCHITECTURES

All methods shared the same backbone architecture. For the core contribution of our work, we used Conv4 architecture consisting of 4 convolutional layers with 64 channels (padding 1), interleaved by

batch normalization (Ioffe & Szegedy, 2015), ReLU activation function, and max-pooling (kernel size 2, and stride 2) (Chen et al., 2019). Relation Network used max-pooling only for the last 2 layers of the backbone to account for the relation module. The relation module consisted of two additional convolutional layers, each followed by batch norm, ReLU, max-pooling).

A.3 Datasets and Distributions

We meta-trained methods on a variety of meta-training datasets and distributions. Specifically, we used Mini-ImageNet (Ravi & Larochelle, 2016; Vinyals et al., 2017), CUB-2011 (Wah et al., 2011), and ImageNet Russakovsky et al. (2015). Our main set of experiments controls imbalance of the meta-training set of Mini-ImageNet (Ravi & Larochelle, 2016) ($\mathcal{D}_{train}$). In order to estimate how significant dataset imbalance can be, we have to eliminate other factors that could influence the meta-learning performance. For this reason, and to have enough samples for majority classes, we halve of the total number of samples from the original $\mathcal{D}_{train}$ of Mini-ImageNet, and denote this halved dataset as $\mathcal{D}'_{train}$, where $|\mathcal{D}'_{train}| \approx 64 \times 300 = 19200$. We induce imbalance into the dataset according to one of the $\mathcal{I}$-distributions as outlined Section 3. For simplicity, we do not modify the distribution in the meta-validation and meta-testing datasets ($\mathcal{D}_{val}$ and $\mathcal{D}_{test}$), and we keep them the same as in the original Min-ImageNet. To emulate a stronger domain-shift scenario, we also evaluate trained models on 50 randomly selected classes from CUB-2011 (Wah et al., 2011). In Appendix C.1 we explore reducing the size of the meta-training dataset to contain $|\mathcal{D}''_{train}| \approx 32 \times 150 = 4800$. In Appendix C.2, we induce long-tail imbalance in ImageNet (Russakovsky et al., 2015). More details can be found in the appropriate sections.

A.4 Meta-Training / Pre-Training

All methods follow a similar three-phase learning procedure: meta-training, meta-validation, and meta-testing. During meta-training, an FSL model was exposed to 50k tasks sampled from $\mathcal{D}_{train}$ or imbalanced variants. After every 500 tasks, the model was validated on tasks from $\mathcal{D}_{val}$ and the best performing model was updated. At the end of the meta-training phase, the best model was evaluated on tasks sampled from $\mathcal{D}_{test}$. SimpleShot follows a similar three-phase procedure but with the meta-training phases exchanged for conventional pre-training on mini-batches of size 128. In all three meta-phases, we used 15 query samples per class, except for the 20-way Prototypical Network, where we used 5 query samples per class during meta-training to allow for a higher number of samples in the support set. All methods were meta-validated on 200 tasks/mini-batches every 250 meta-training tasks/mini-batches to select the best performing model. We used a learning rate of 1×10^{-3} for the first 12.5k tasks, then reduced it to 1×10^{-4} for the remaining tasks.

Data Augmentation. During the meta-/pre-training phases, we apply standard data augmentation techniques, following a similar setup to Chen et al. (2019), with a random rotation of 10 degrees, scaling, random color/contrast/brightness jitter. Meta-validation and meta-testing had no augmentation. All images are resized to 84 by 84 pixels.

A.5 Meta-Testing

The final test performances were measured on a random sample of 600 tasks. For all evaluations, we used standard, $balanced$ 5-shot 5-way tasks. We report the average one standard deviation in brackets and error-bars.

B VERIFICATION OF IMPLEMENTATION

We implement the FSL methods in PyTorch, adapting the implementation of (Chen et al., 2019) but also borrowing from other implementations online (see individual method files in the source code for individual attribution). However, we have heavily modified these implementations to fit our imbalanced FSL framework, which also offers standard and continual FSL compatibility (Antoniou et al., 2020). We provide our implementations for ProtoMAML and BTAML for which no open-source implementation in PyTorch existed. To verify our implementations, we compare methods on the standard balanced 5-shot 5-way task with reported accuracy. Results are presented in Table 2. We see that algorithms achieve very similar performance with no less than 3% accuracy points compared to the reported performance.

Table 2: Results of standard 5-shot 5-way experiments on Mini-ImageNet as achieved with our implementation compared to the original (reported) accuracy and other work. Other Sources's Accuracies were taken from: * (Chen et al., 2019), † (Snell & Zemel, 2020), ‡ (Vogelbaum et al., 2020)

Model	Acc (95%CI)	Original Acc(95%CI)	Other Sources Acc(95%CI)
Baseline (fine-tune) (Chen et al., 2019)	$62.67_{\pm 0.70}$	$62.53_{\pm 0.69}$	-
Baseline++ (Chen et al., 2019)	$\mathbf{66.43}_{\pm \mathbf{0.66}}$	$66.43_{\pm 0.63}$	-
Matching Net (Vinyals et al., 2017)	$62.27_{\pm 0.69}$	$55.31_{\pm 0.73}$	$63.48_{\pm 0.66}$ *
ProtoNet (Snell et al., 2017)	$64.37_{\pm 0.71}$	$65.77_{\pm 0.70}$	$64.24_{\pm 0.72}$ *
ProtoNet (20-way) (Snell et al., 2017)	$65.76_{\pm 0.70}$	$\mathbf{68.20}_{\pm \mathbf{0.66}}$	$\mathbf{66.68}_{\pm \mathbf{0.68}}$ *
Relation Net (CE) (Sung et al., 2017)	$64.76_{\pm 0.68}$	$65.32_{\pm 0.70}$	$66.60_{\pm 0.69}$ *
DKT (Patacchiola et al., 2020)	$62.92_{\pm 0.67}$	$64.00_{\pm 0.09}$	$62.88_{\pm 0.46}$ †
SimpleShot (Wang et al., 2019)	$63.74_{\pm 0.69}$	$66.92_{\pm 0.17}$	-
MAML (Finn et al., 2017)	$61.83_{\pm 0.71}$	$63.15_{\pm 0.91}$	$62.71_{\pm 0.71}$ *
ProtoMAML (Triantafillou et al., 2020)	$65.87_{\pm 0.71}$	-	$60.70_{\pm 0.99}$ ‡

C ADDITIONAL EXPERIMENTS.

C.1 SMALL AND IMBALANCED META-TRAINING DATASET

In this section, we examine whether the effect of imbalance could be influenced by a smaller dataset size. Specifically, we construct a new set of datasets denoted by $\mathcal{D}''_{train}$ containing 1/8th of samples in $\mathcal{D}_{train}$ of Mini-ImageNet, and 32 classes selected uniformly at random, $|\mathcal{D}''_{train}| = 4800$. Table 3 are a break-down of Figure 2. Overall, the performance drops as the number of minority classes increases but the effect still remains quite low ($< 3\%$ absolute difference).

Table 3: Evaluation accuracy after meta-training on $\mathcal{D}''_{train}$ derived from Mini-ImageNet with various imbalance distributions. Small differences in accuracy between $balanced$ and other distributions, suggest a small effect of imbalance at dataset level. *Left:* Evaluation on the meta-testing dataset of Mini-ImageNet. *Right:* Evaluation on the meta-testing dataset of CUB.

$\mathcal{D}''_{train} \rightarrow \mathcal{D}_{test}$	Mini-ImageNet → Mini-ImageNet				Mini-ImageNet → CUB			
Imbalance in $\mathcal{D}''_{train}$	$balanced$ (150, 150, 32, -)	$step$-8 (30, 190, 32, 8)	$step$-16 (30, 270, 32, 16)	$step$-24 (30, 510, 32, 24)	$balanced$ (150, 150, 32, -)	$step$-8 (30, 190, 32, 8)	$step$-16 (30, 270, 32, 16)	$step$-24 (30, 510, 32, 24)
Baseline (fine-tune)	56.07$\pm$0.71	56.19$\pm$0.72	56.17$\pm$0.71	55.81$\pm$0.72	55.80$\pm$0.68	55.19$\pm$0.70	55.14$\pm$0.70	53.38$\pm$0.69
Baseline++	54.35$\pm$0.67	54.24$\pm$0.67	53.90$\pm$0.67	53.25$\pm$0.68	50.72$\pm$0.69	50.64$\pm$0.70	50.44$\pm$0.69	49.50$\pm$0.67
Matching Net	55.79$\pm$0.68	54.76$\pm$0.69	53.44$\pm$0.68	51.54$\pm$0.67	52.27$\pm$0.71	50.44$\pm$0.72	49.61$\pm$0.72	49.60$\pm$0.71
ProtoNet	55.90$\pm$0.71	54.71$\pm$0.70	54.75$\pm$0.71	53.75$\pm$0.71	51.24$\pm$0.73	50.93$\pm$0.73	50.96$\pm$0.72	49.47$\pm$0.73
Relation Net (CE)	55.41$\pm$0.70	53.80$\pm$0.69	51.15$\pm$0.70	50.24$\pm$0.68	51.42$\pm$0.70	50.06$\pm$0.72	49.12$\pm$0.71	47.12$\pm$0.70
DKT	57.21$\pm$0.67	56.23$\pm$0.68	54.65$\pm$0.67	53.65$\pm$0.69	**56.36$\pm$0.69**	54.76$\pm$0.68	54.69$\pm$0.69	**54.98$\pm$0.70**
SimpleShot	**58.80$\pm$0.74**	**58.61$\pm$0.74**	**58.46$\pm$0.75**	**58.05$\pm$0.75**	56.04$\pm$0.71	**55.73$\pm$0.72**	**56.01$\pm$0.71**	53.55$\pm$0.72
MAML	53.17$\pm$0.75	53.16$\pm$0.74	52.64$\pm$0.75	51.79$\pm$0.73	53.56$\pm$0.73	52.90$\pm$0.76	52.50$\pm$0.72	53.13$\pm$0.73
ProtoMAML	57.75$\pm$0.71	56.74$\pm$0.71	56.05$\pm$0.71	54.34$\pm$0.70	54.85$\pm$0.72	54.02$\pm$0.75	53.16$\pm$0.71	53.20$\pm$0.71
Avr. Diff. to $balanced$	-	0.7	-1.5	-2.4	-	-0.8	-1.2	-2.0

C.2 LONG-TAIL IMBALANCE

In this section, we explore the performance under a larger imbalance setting with long-tail distribution (Salakhutdinov et al., 2011). We induce a long-tail distribution on classes from ImageNet (Russakovsky et al., 2015). We used ResNet-10 as the backbone. We partitioned ImageNet to contain 900 classes for $\mathcal{D}_{train}$ distributed according to $balanced$ or $long\text{-}tail$, while $\mathcal{D}_{val}$ and $\mathcal{D}_{test}$ contained 50 classes each with 500 randomly selected samples per class. The set of classes in each dataset is kept the same for all experiment repeats. We induce $long\text{-}tail$ imbalance in $\mathcal{D}_{train}$ using the Power-Law distribution (Reed, 2001) with a power of 10, a minimum of 20 samples per class (to allow 5-shot 15-query task), and a maximum of 1300 samples per class possible for ImageNet. This means that $\rho = 65$ and the top 20% majority classes in distribution account for 80% of all data points in $\mathcal{D}_{train}$. We shuffled the classes within $\mathcal{D}_{train}$ such that the number of samples for a particular class could vary between repeats. The Balanced dataset contained samples distributed uniformly among the 900 classes (i.e. 137 samples per class) such that $|\mathcal{D}_{train}| \approx 123300$ for both distributions (within 500 samples difference between them). Table 4 shows the accuracy performance after training on 1800 balanced tasks across 3 seeds. The results show that meta-learner models – ProtoNet, MAML, ProtoMAML– are particularly susceptible to the larger imbalance, experiencing -6.3%, -4.1%, -4.5% drop in accuracy, respectively. Interestingly, the Baseline seems to be the least affected by the imbalance with a non-significant drop in performance (-0.1%).

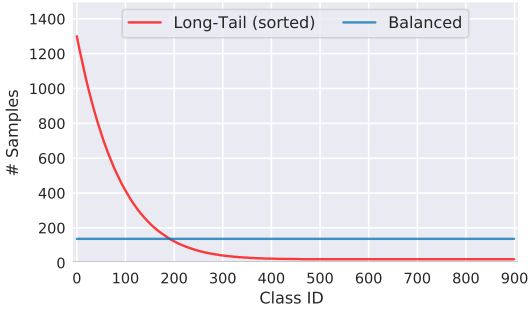

Figure 3: Figure showing the class distribution for Balanced vs Long-Tail (sorted by class size).

Table 4: Accuracy performance on Long-Tail distribution vs balanced with approximately the same number of samples in total.

I in $\mathcal{D}_{train}$	$balanced$ (137, 137, 900,-)	$long\text{-}tail$ (20, 1300, 900,-)
Baseline	62.57$\pm$0.85	62.47$\pm$0.85
MAML	62.83$\pm$0.90	58.69$\pm$0.90
ProtoNet	67.97$\pm$0.83	61.67$\pm$0.86
ProtoMAML	65.44$\pm$0.88	60.97$\pm$0.89